Nonparametric methods and `tidyr`

BIO5312 FALL2017

STEPHANIE J. SPIELMAN, PHD

General notes

Results means the literal results of the test

- **Value of the test statistic**
- **P-value**
- Estimate, CI

Conclusions means our interpretation of those results

- **If P > alpha**
  - **Fail to reject Ho, no evidence in favor of Ha**
- **If P <= alpha,**
  - **Reject Ho, found evidence in favor of Ha, make directional conclusion if possible**

Our bag of tests

Numeric data: *t*-tests

- One sample/paired
- Two sample

Categorical data

- One categorical variable with two levels: Binomial
- One categorical variable with >two levels: Chi-squared goodness of fit
- Two categorical variables: Contingency table
  - Chi-squared for large samples
  - Fisher's exact test for small samples

Nonparametric tests

Make no* assumptions about how your samples are distributed

- Also known as *distribution-free* tests

Lower *false positive* rate than parametric methods when assumptions not met

Less powerful than parametric methods

Used primarily when sample sizes are small or non-normal (for a *t*-test)

Our new bag of tests

One sample or paired *t*-test

- Sign test
- Wilcoxon signed-rank test

Two sample *t*-test

- Mann Whitney *U*-test (Wilcoxon rank sum test)

Many nonparametric tests are based on data ranks

X	Ranks
10.8	4
13.5	6
9.1	3
11.5	5
15.7	7
4.3	1
8.4	2

The sign test for single numeric samples

H_0: The median of a sample is equal to <null median>

H_A: The median of a sample is not equal to <null median>

Procedure:

- Determine your null median
- Assign each value in your sample as + or - if above or below median
- Test whether there are same number of +, -

Example: Sign test

An environmental biologist measured the pH of rainwater on 7 different days in Washington state and wants to know if rainwater in the region can be considered acidic (< pH 5.2).

pH	Sign
4.73	-
5.28	+
5.06	-
5.16	-
5.25	+
5.11	-
4.79	-

5+
2-

The sign test is a binomial test with p=0.5

H_0: The median pH of WA rain is 5.2.

H_A: The median pH of WA rain is less then 5.2

```
> binom.test(2, 7, 0.5, alternative = "less")
Exact binomial test

data:  2 and 7
number of successes = 2, number of trials = 7, p-value = 0.4531
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.03669257 0.70957914
sample estimates:
probability of success
             0.2857143
```

Results and conclusions

Our test gave P=0.4531. This is greater than 0.05 so we **fail to reject** the null hypothesis. We have **no evidence that** rainwater in WA state is acidic.

Sign test in R

```
rain <- tibble(pH = c(4.73, 5.28, 5.06, 5.16, 5.25, 5.11, 4.79))

rain %>% mutate(sign = sign(5.2 - pH))
         pH  sign
      <dbl> <dbl>
    1  4.73     1
    2  5.28    -1
    3  5.06     1
    4  5.16     1
    5  5.25    -1
    6  5.11     1
    7  4.79     1

rain %>% mutate(sign = sign(5.2 - pH)) %>% group_by(sign) %>% tally()
       sign     n
      <dbl> <int>
    1    -1     2
    2     1     5
```

See one, do one

Wilcoxon signed-rank test

Updated version of sign test that also considers magnitude

pH	Sign
4.73	-
5.28	+
5.06	-
5.16	-
5.25	+
5.11	-
4.79	-

Adding ranks to the procedure

H_0: The median pH of WA rain is 5.2.

H_A: The median pH of WA rain is not then 5.2

pH	**Sign**	**|*x* – null|**	**rank**
4.73	-1	0.47	7
5.28	1	0.08	3
5.06	-1	0.14	5
5.16	-1	0.04	1
5.25	1	0.05	2
5.11	-1	0.09	4
4.79	-1	0.41	6

Compute the test statistic **W (R)**

W = min(sum negative sign ranks, sum positive sign ranks)

Negative sign ranks:

- 7+5+1+4+6 = **23**

Positive sign ranks:

- 3+2 = **5**

```
### Two sided P-value ###
### psignrank(w, n) ###
> 2*psignrank(5,7)
[1] 0.15625
```

Sign	rank
-1	7
1	3
-1	5
-1	1
1	2
-1	4
-1	6

Wilcoxon signed-rank, the long way

```
> rain %>% mutate(sign = sign(5.2 - pH), rank = rank(abs(5.2 - pH)))
     pH  sign  rank
   <dbl> <dbl> <dbl>
1  4.73     1     7
2  5.28    -1     3
3  5.06     1     5
4  5.16     1     1
5  5.25    -1     2
6  5.11     1     4
7  4.79     1     6

> rain %>% mutate(sign = sign(5.2 - pH), rank = rank(abs(5.2 - pH))) %>%
group_by(sign) %>% summarize(sum(rank))
    sign `sum(rank)`
   <dbl>       <dbl>
1    -1           5
2     1          23

> psignrank(5, nrow(rain))
  [1] 0.078125
```

Wilcoxon signed-rank, the obvious way

```
> rain <- tibble(pH = c(4.73, 5.28, 5.06, 5.16, 5.25, 5.11,
4.79))

> wilcox.test(rain$pH, mu = 5.2)
Wilcoxon signed rank test


data:  rain$pH
V = 5, p-value = 0.1563
alternative hypothesis: true location is not equal to 5.2
```

Wilcoxon signed-rank is not foolproof

Although nonparametric, assumes population are symmetric around the median (no skew)

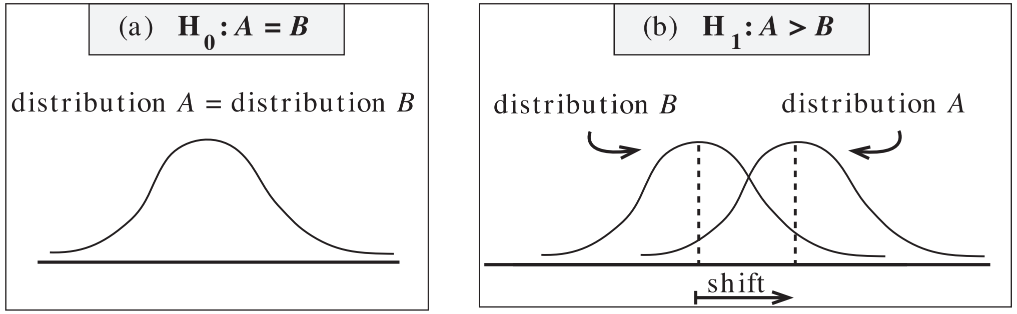
This is hard to meet, so recommendation is to use the sign test.

See one, do one

Mann-Whitney U test (aka Wilcoxon rank sum)

Nonparametric test to compare two numeric samples

Assumes samples have the same shape and detects a *shift* between distributions.


(a) $\mathbf{H_0}$: $A = B$
distribution A = distribution B
(b) $\mathbf{H_1}$: $A > B$
distribution B
distribution A
shift

H_0: Sample 1 and sample 2 have the same underlying distribution location.
H_A: Sample 1 and sample 2 have different (>/<) underlying distribution location.

The tedious steps to MW-U test

1. Pool the data and rank everything

2. Sum ranks for group 1 and group 2 each → R_1 and R_2

3. Compute *U* statistic as $\min(U_1,U_2)$ from ranks:

   - $U_1 = R_1 - \frac{n_1(n_1+1)}{2}$
   - $U_1 + U_2 = n_1 n_2$

4. Get the pvalue in R: `pwilcox(U, n₁, n₂)`

Minimal example

Sample 1: 8, 15, 17

Sample 2: 22, 10, 16, 28

R1 = 1+3+5 = 9

R2 = 2+4+6+7 = 19

$$U_1 = R_1 - [n_1(n_1+1)/2]$$
$$= 9 - [3(4)/2] = 3$$

$$U_2 = n_1 n_2 - U_1$$
$$= 3*4 - 3 = 9$$

```
### One tailed P ###
> pwilcox(3, 3, 4)
[1] 0.2
```

8	1
10	2
15	3
16	4
17	5
22	6
28	7

Minimal example… in R

```
> wilcox.test(c(8, 15, 17), c(22, 10, 16, 28))


Wilcoxon rank sum test


data:  c(8, 15, 17) and c(22, 10, 16, 28)
W = 3, p-value = 0.4
alternative hypothesis: true location shift is not equal to 0
```

Major caveat: ties in data

Test assumes all data is **ordinal**

Sample 1: 8, 15, 17

Sample 2: 22, 10, 16, 17

Assign all values in tie the **average** rank

8	1
10	2
15	3
16	4
17	5.5
17	5.5
22	7

Example in R, with ties

```
> wilcox.test(c(8, 15, 17), c(22, 10, 16, 17))

Wilcoxon rank sum test with continuity correction

data:  c(8, 15, 17) and c(22, 10, 16, 17)
W = 3.5, p-value = 0.4755
alternative hypothesis: true location shift is not equal to 0


Warning message:
In wilcox.test.default(c(8, 15, 17), c(22, 10, 16, 17)) :
  cannot compute exact p-value with ties
```

See one, do one

What is a dataset?

A collection of **values**

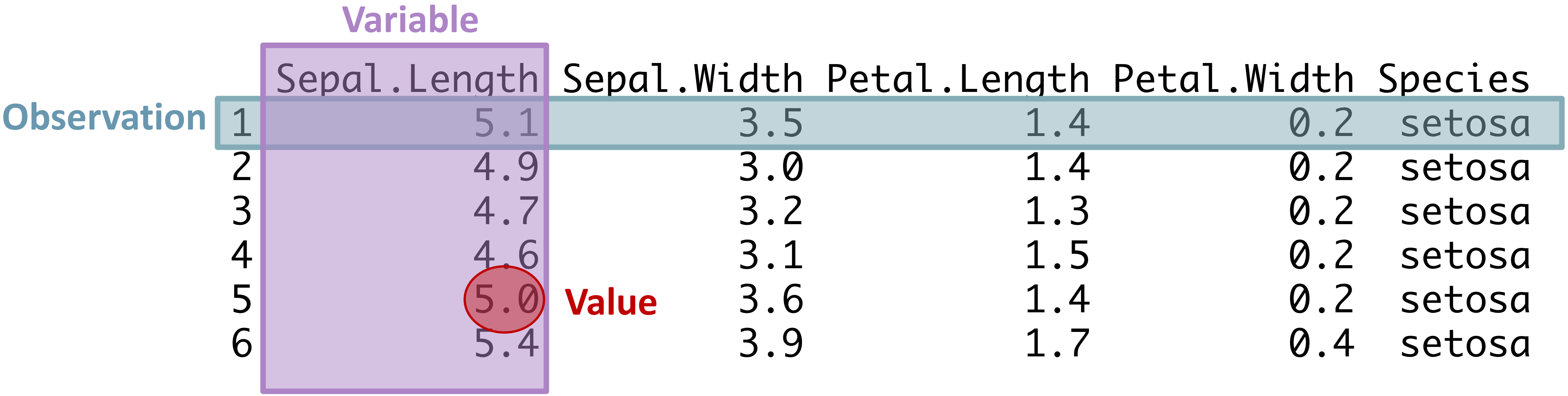
Each **value** belongs to a **variable** and an **observation**

Variables contain all values that measure the same underlying attribute ("thing")

Observations contain all values measured on the same unit across attributes.

Hadley Wickham
https://cran.r-project.org/web/packages/tidyr/vignettes/tidy-data.html

The iris dataset (what else?)



```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

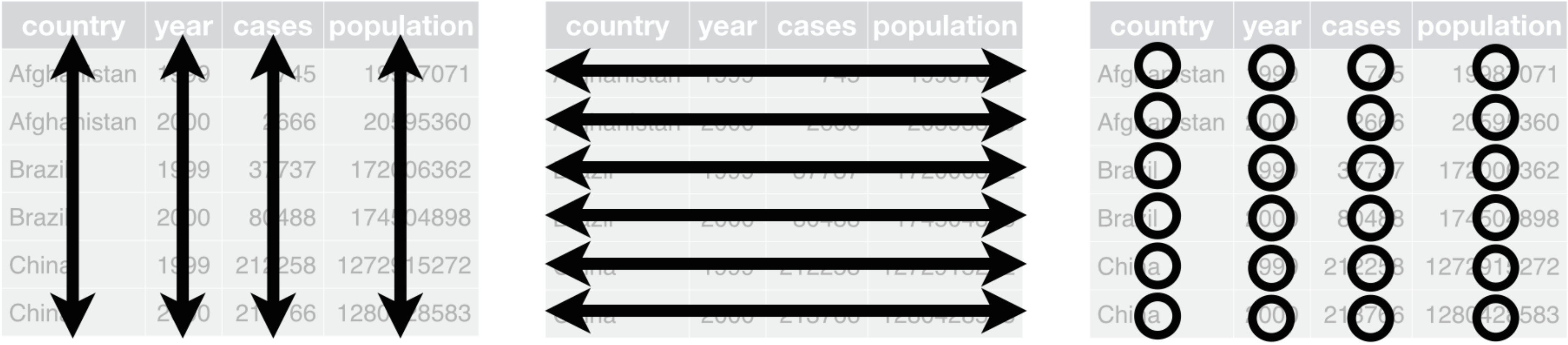
This is a **tidy dataset**

Each variable forms a column.

Each observation forms a row.

Each type of observational unit forms a table.

Tidy data provides a consistent approach to data management that greatly facilitates downstream analysis and viz

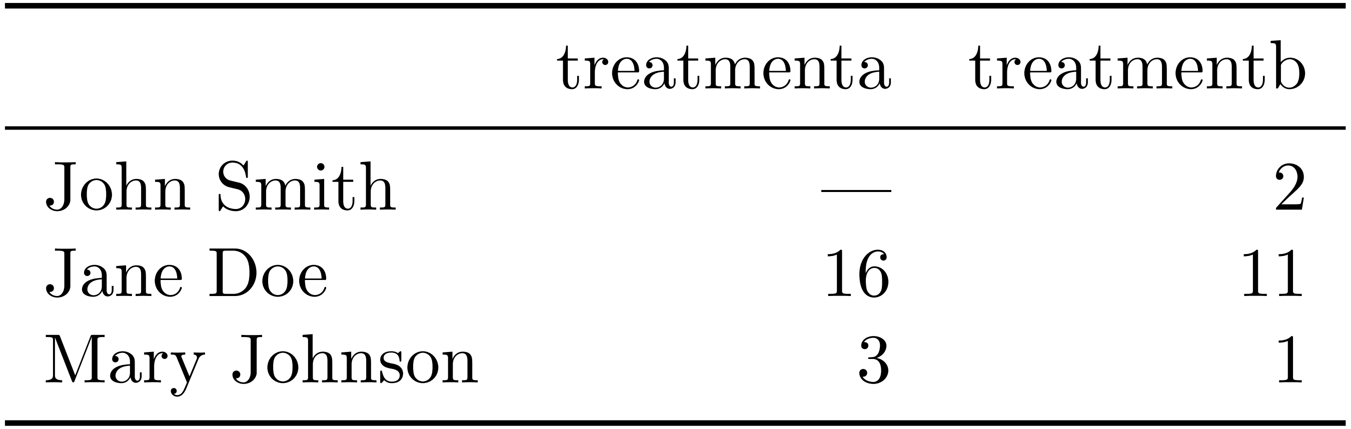


variables

observations

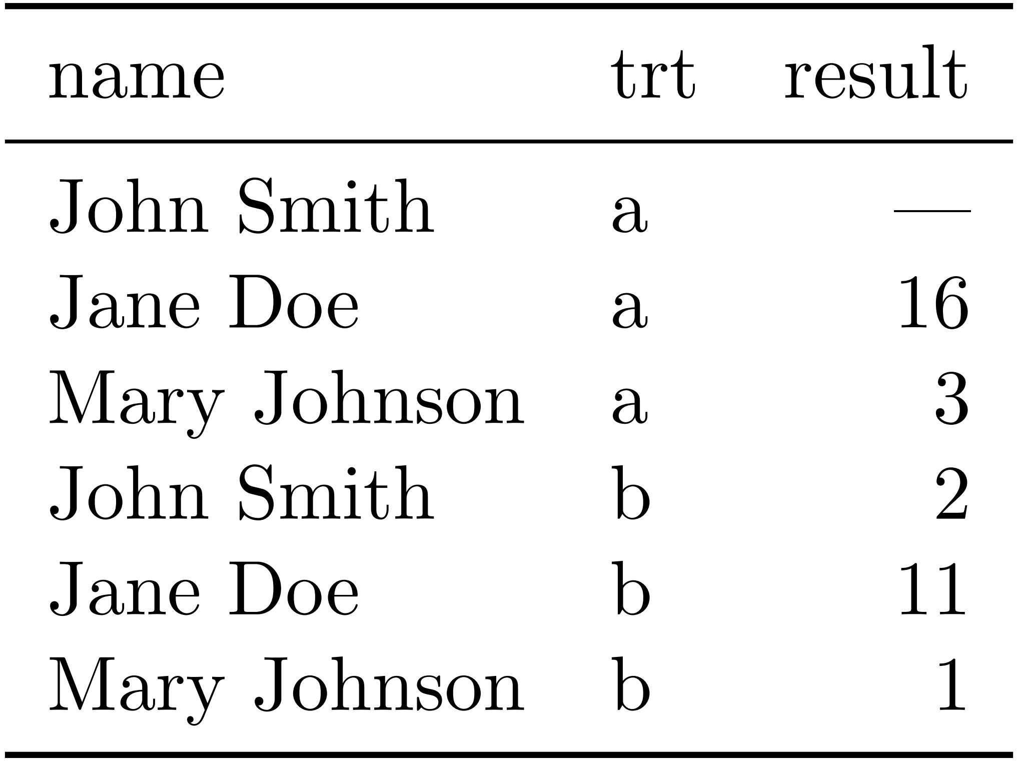
values

Messy vs tidy data



	treatmenta	treatmentb
John Smith	—	2
Jane Doe	16	11
Mary Johnson	3	1

What are the **variables** in this data?
What are the **observations** in this data?



name	trt	result
John Smith	a	—
Jane Doe	a	16
Mary Johnson	a	3
John Smith	b	2
Jane Doe	b	11
Mary Johnson	b	1

Do it yourself: Convert to tidy data

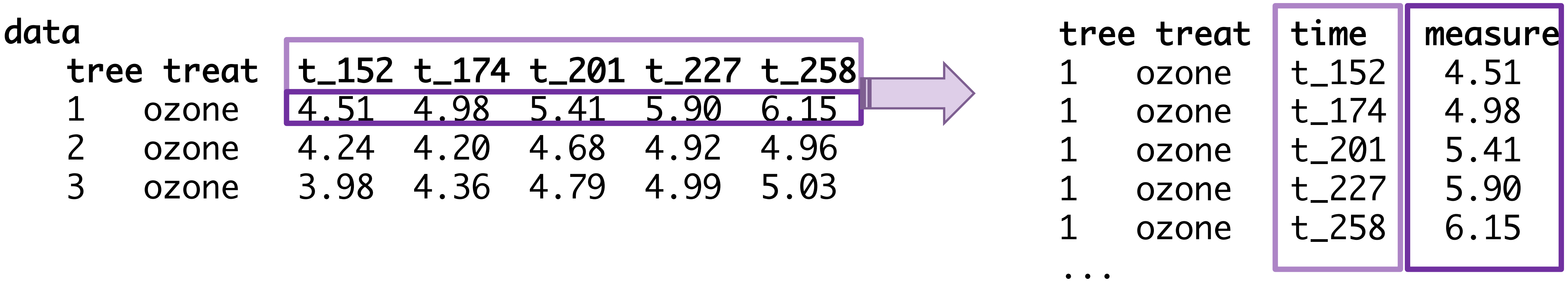
	survived	died
drug	15	3
placebo	4	11

treatment	outcome	count
drug	survived	15
placebo	survived	4
drug	died	3
placebo	died	11

The fundamental verbs of `tidyr`

`gather()`	Gather multiple columns into key:value pairs
`spread()`	Spread key:value pairs over multiple columns
`separate()`	Separate columns
`unite()`	Join columns

gather() makes wide tables narrow



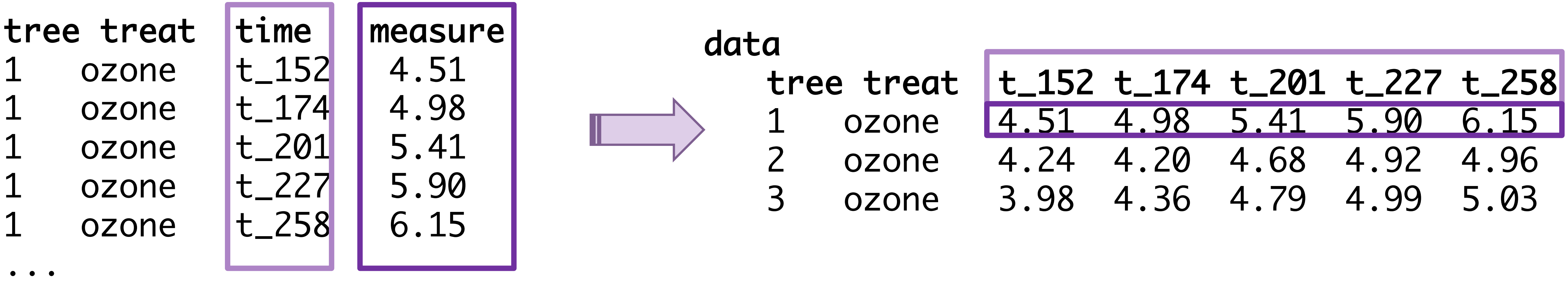
```
data
    tree treat  t_152 t_174 t_201 t_227 t_258
    1    ozone  4.51  4.98  5.41  5.90  6.15
    2    ozone  4.24  4.20  4.68  4.92  4.96
    3    ozone  3.98  4.36  4.79  4.99  5.03
```

```
tree treat  time   measure
1    ozone  t_152  4.51
1    ozone  t_174  4.98
1    ozone  t_201  5.41
1    ozone  t_227  5.90
1    ozone  t_258  6.15
...
```

```
data %>% gather(time, measure, t_152:t_258)
```

KEY **VALUE**

spread() makes narrow tables wide



```
tree treat  time   measure
1    ozone  t_152   4.51
1    ozone  t_174   4.98
1    ozone  t_201   5.41
1    ozone  t_227   5.90
1    ozone  t_258   6.15
...
```

```
data
    tree treat  t_152 t_174 t_201 t_227 t_258
    1    ozone  4.51  4.98  5.41  5.90  6.15
    2    ozone  4.24  4.20  4.68  4.92  4.96
    3    ozone  3.98  4.36  4.79  4.99  5.03
```

```
data %>% spread(time, measure)
```

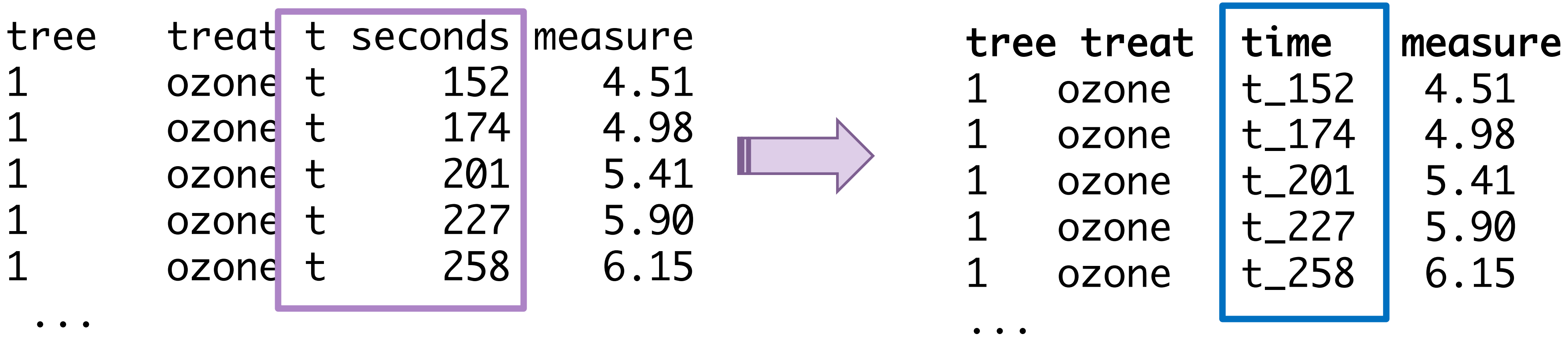
separate() separates columns

```
tree treat   time   measure
1    ozone   t_152   4.51
1    ozone   t_174   4.98
1    ozone   t_201   5.41
1    ozone   t_227   5.90
1    ozone   t_258   6.15
...
```

```
tree     treat t  seconds measure
1        ozone t      152    4.51
1        ozone t      174    4.98
1        ozone t      201    5.41
1        ozone t      227    5.90
1        ozone t      258    6.15
  ...
```

```
data %>% separate(time, into=c("t", "seconds"), sep = "_")
```

unite() unites columns

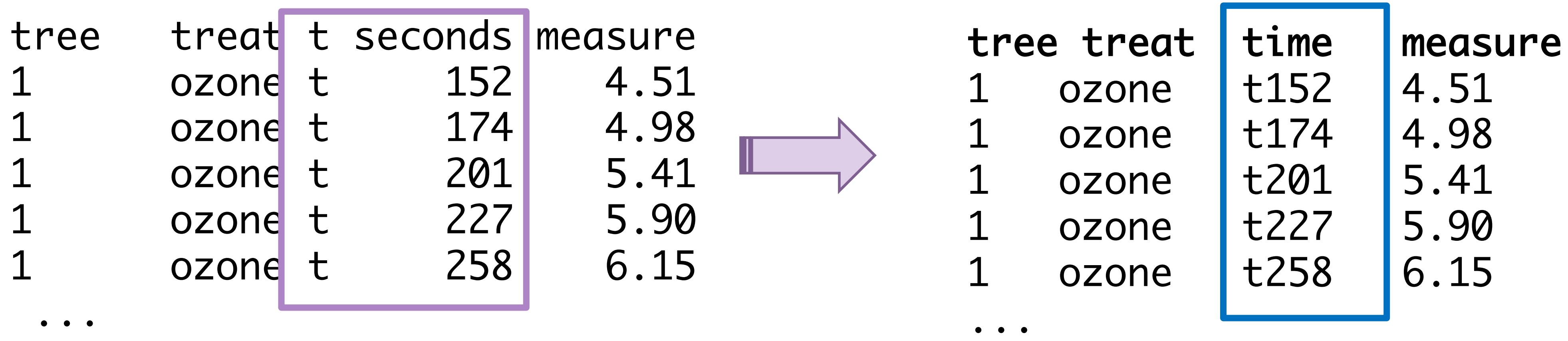


tree	treat	t	seconds	measure
1	ozone	t	152	4.51
1	ozone	t	174	4.98
1	ozone	t	201	5.41
1	ozone	t	227	5.90
1	ozone	t	258	6.15
...				

tree	treat	time	measure
1	ozone	t_152	4.51
1	ozone	t_174	4.98
1	ozone	t_201	5.41
1	ozone	t_227	5.90
1	ozone	t_258	6.15
...			

```
data %>% unite(time, t, seconds)
```

unite() unites columns



```
tree    treat t seconds measure
1       ozone t     152    4.51
1       ozone t     174    4.98
1       ozone t     201    5.41
1       ozone t     227    5.90
1       ozone t     258    6.15
 ...
```

```
tree treat  time  measure
1    ozone  t152  4.51
1    ozone  t174  4.98
1    ozone  t201  5.41
1    ozone  t227  5.90
1    ozone  t258  6.15
...
```

```
data %>% unite(time, t, seconds, sep = "" )
```